

# Natural Recording Utility

With the Natural Recording utility, you can record a Natural session and later play back the recorded session.

**Note:**

While the Natural Recording utility is used to record input data and keyboard actions, the NATPAGE utility (see the relevant section) is used to solely record output data as described in the relevant documentation.

Related Documentation: Terminal Commands, Natural Programming Reference.

This section covers the following topics:

- Purpose of Recording
  - Data and Functions Recorded
  - Recording a Session
  - Playing Back a Recording
  - Manipulating a Recording
- 

## Purpose of Recording

The Natural Recording utility can be used for the following purposes:

- **Demonstration**  
Instead of having to type in several commands, such as input data by hand, you can play back a recorded sequence of keyboard actions to demonstrate a standard procedure.
- **Application development**  
When applying the same modifications to several objects (for example, programs or maps), you can use a recording to reduce the amount of work involved and at the same time ensure that the modifications are actually the same for all objects affected.
- **Testing**  
You can execute a standard testing procedure by simply playing back a recording.
- **Quality control**  
Before and after making changes to an application, you can play back a recording and compare the results of the two runs to make sure that certain things were not affected by the changes.
- **User training**  
You can incorporate the playback of recordings into training programs for users, to show them specific procedures. Also, you can record users' keyboard actions in a session and then inform them of any errors they make or of ways to carry out actions more efficiently. The recording of users' actions can also help you to detect any flaws in an application's user interface.

## Data and Functions Recorded

The Natural Recording utility records:

- All input data and commands (including terminal commands) entered on the screen,
- Any function keys pressed.
- The current cursor position as contained in the Natural system variable \*CURSOR.

## Recording a Session

Below are the steps required to activate and deactivate a recording:

- Specifying Libraries
- Activating a Recording
- Deactivating a Recording

### Specifying Libraries

#### To specify the library in which all subsequent recordings are to be stored

- Enter the terminal command **%B=library-name**.

If you activate the recording process without having specified *library-name*, the name of the library in which the recording is stored is the same as the value of the system variable \*INIT-USER at the time when the recording process is activated.

When you log on to another library during a session being recorded, the library in which the recording is being stored remains the same (that is, either the one specified with **%B=** or the \*INIT-USER library); this means that one recording can record keyboard actions across multiple applications.

### Activating a Recording

#### To activate a recording

- Enter the terminal command **%Bname**.

All subsequent keyboard actions are recorded.

*Name* denotes the name under which the data recorded are saved in source form as a Natural object of the type Recording. You can treat this source as any other Natural source (for example, delete it, copy it), except that you must not edit it: recordings contain binary data an editor will destroy.

*Name* can only be specified once. If a recording object of the same name already exists in the library specified for recording, Natural returns the message "Error in recording activation".

#### **Attention:**

Any situation that leads to a backout transaction or rollback (for example, a non-activity timeout) while a recording is in progress, will delete part of the recording thus making the entire recording useless.

Terminal Command **%Aname** included in a recording should be followed by Terminal Command **%B** as described in Recording %A below.

### Deactivating a Recording

#### To deactivate a recording

- Enter the terminal command **%B**.

The recording has terminated.

## Playing Back a Recording

When a recording is played back, the sequence of, for example, commands and function keys is actually executed again.

The recording is independent of the terminal type; that is, a session recorded on one terminal can be played back on a terminal of another type. You can also play back a recording in batch mode; a recorded online session may, of course, react differently when played back in batch.

Below is information on:

- Step Mode and Background Mode
- Activating a Playback
- Interrupting a Playback

### Step Mode and Background Mode

A recording can be played back in two modes: background mode and step mode.

In background mode, the entire recording is played back invisibly; that is, all keyboard actions of the recording are carried out without anything being displayed to you on the terminal screen during the execution of the recording. You cannot interrupt a recording that is played back in background mode, unless the recording contains the terminal command `%R` as explained below.

In step mode, a recording is played back step by step and all keyboard actions are displayed on the screen. By choosing any function key, you proceed from one step to the next. In step mode, it is also possible for you to interrupt the recording by pressing `CLEAR` as explained below.

By default, a recording is played back in background mode.

#### To set modes

- Enter the terminal command `%GON` to activate step mode.
- Enter the terminal command `%GOFF` to deactivate step mode and activate background mode.
- Enter the terminal command `%G` to toggle between step and background mode.

### Activating a Playback

#### To play back a recording

- Enter the Natural terminal command `%Aname`.

The recording saved under the specified name is executed again.

#### **Recording `%Aname`**

If you issue the command `%Aname` while a session is being recorded, the recording specified with `%Aname` is not executed but the command `%Aname` is included into the object source that is being recorded. Thus, you can execute a recording from within another recording and concatenate a series of recording to one another. However, you cannot have nested recordings; the execution of the recording that contains the `%Aname` command stops after this command and is not resumed when the execution of *name* finishes. As a result, the data recorded after `%Aname` will never be played back. To avoid this, you should enter `%B` immediately after you have entered `%Aname` in a recording.

## Interrupting a Playback

### To interrupt a recording that is played back in step mode

- Press CLEAR.

Once you have interrupted a recording, you have the following options:

- You can continue your session normally from the point where you stopped the recording.
- You can insert additional keyboard actions into the recording: after you have pressed CLEAR, enter the command **%B** and all actions you perform are inserted into the source of the recording until you enter **%B** again. Then, the execution of the recording is resumed.
- You can alter the next step in the recording: after you have pressed CLEAR, enter the command **%R**, then enter the input data for the next step. The newly entered input data overwrite the input data for this step in the recorded source. When you press ENTER, this step is executed with the new input data and subsequently the execution of the recording is resumed.
- You can execute any helproutine: after you have pressed CLEAR, enter the command **%J** directly followed by the name of the desired helproutine. The helproutine is invoked and the execution of the recording is continued as soon as the execution of the helproutine ends.

## Manipulating a Recording

By recording the terminal command **%R**, you can manipulate a single step in a recording when it is played back. This applies in step mode and in background mode. In background mode, **%R** is the only way to interact with a recording that is being played back. Interaction, for example, may be required to provide an input option for sensitive data, such as passwords which are unknown at the time of the recording.

If the terminal command **%R** (redisplay last screen) has been recorded, the subsequent screen is open for user input when the recording is played back; that is, the input data for this screen are not taken from the recording but from what the user enters. Subsequently, the execution of the recording is continued.